Digital Watermarking of Raw and Compressed Video

Frank Hartung

Bernd Girod

Telecommunications Institute
University of Erlangen-Nuremberg
Cauerstrasse 7, 91058 Erlangen, Germany
{hartung, girod}@nt.e-technik.uni-erlangen.de

ABSTRACT

Embedding information into multimedia data is a topic that has gained increasing attention recently. For video broadcast applications, watermarking of video, and especially of already encoded video, is interesting. We present a scheme for robust interoperable watermarking of MPEG-2 encoded video. The watermark is embedded either into the uncoded video or into the MPEG-2 bitstream, and can be retrieved from the decoded video. The scheme working on encoded video is of much lower complexity than a complete decoding process followed by watermarking in the pixel domain and re-encoding. Although an existing MPEG-2 bitstream is partly altered, the scheme avoids drift problems. The scheme has been implemented and practical results show that a robust watermark can be embedded into MPEG encoded video which can be used to transmit arbitrary binary information at a data rate of several bytes/second.

Keywords: watermarking, video, MPEG-2, video broadcast

1 Introduction

With digital broadcast of video, legal issues of copyright protection have become more important, since the inherent decrease of quality of analog video duplication has vanished in digital applications. A favorable method of copyright protection is digital watermarking of the multimedia data, i.e., adding a "watermark" (in other publications also called "label", "tag" or "signature") that authenticates the legal copyright holder and that cannot be manipulated or removed without, at the same time, impairing the multimedia data so much that they are of no commercial value any more.[1-8] Alternatively, an individual watermark might also be included at the conditional access unit in the transmitter that encrypts the video for the individual receiver in order to identify the receiver if he copies and illegally distributes the video, as shown in Fig. 1. While previous publications[1-8] do not deal with watermarking in the bitstream domain of coded video, this is an especially interesting topic. High-quality MPEG encoding is very complex, in some applications it is even done interactively with fine-tuning of parameters by a human operator. Therefore, individual digital watermarking of digital video broadcasted to different receivers can be done only after encoding, but before decoding, as shown in Fig. 1.

0-8194-2356-4/96/$6.00

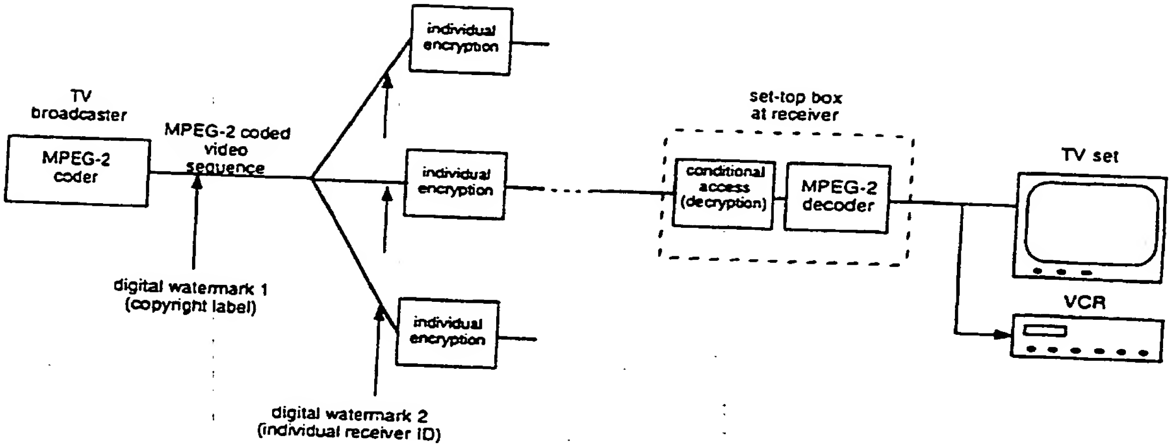
individual encryption
TV broadcaster
MPEG-2 coder
MPEG-2 coded video sequence
individual encryption
set-top box at receiver
conditional access (decryption)
MPEG-2 decoder
TV set
digital watermark 1 (copyright label)
individual encryption
VCR
digital watermark 2 (individual receiver ID)

Figure 1: Transmission scheme for video with individual watermark embedding.

In section 3, we introduce a scheme for spread spectrum like watermarking of uncoded video. In section 4, we show techniques for robust *interoperable* digital watermarking of video where we incorporate the watermark in the bitstream domain of MPEG-2 coded video (that is, without decoding and full re-encoding) and can retrieve it from the decoded video, as shown in Fig. 2. In section 5, possible attacks against watermarks are explained, and remedies are given. In section 6, we present practical results. We have implemented our scheme for watermark of MPEG-2 encoded video which works robust and can embed arbitrary watermark information into encod video at a data-rate of several bytes/second.

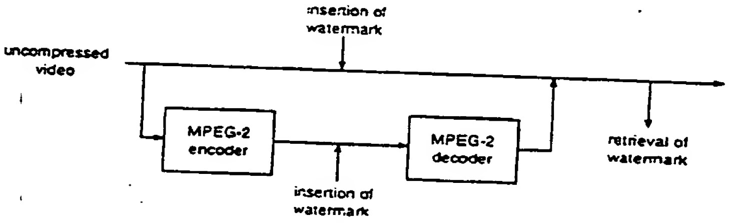

Figure 2: Interoperability of watermarking in the uncoded and coded domain.

2 Requirements on a digital watermarking scheme for video broadcast applications

A digital watermark is a signal carrying information that is embedded into another transport signal, for example into a video signal. A watermarking scheme for video broadcast applications should comply with the following requirements:

- The digital watermark embedded into the video data should be invisible or at least hardly perceptible.

- The watermark should be such that it cannot be removed by intentional or unintentional operations on the bitstream or on the decoded video without, at the same time, degrading the perceived quality of the video so much that it is of no commercial value any more. This requirement is called robustness.
- For broadcast applications, it can be assumed that the broadcaster will usually store the video in compressed format. Therefore, it must be possible to incorporate the watermark into the encoded video, i.e., into the bitstream. It is not feasible to decode and re-encode the video for the purpose of watermarking it.
- Watermarking in the bitstream domain may not increase the bit-rate (at least for constant bit-rate applications). This requirement is not obeyed by previous publications dealing with watermarking of still images during JPEG compression.[2]
- It can be assumed (and it is, in practice, the case) that incorporating a watermark into compressed video has to obey much more constraints than incorporating a watermark into uncompressed video. Therefore, it is advantageous to do so in the domain of uncompressed video wherever possible. Hence, the watermarking algorithm should work interoperable for compressed and uncompressed video with the same type of decoder, that is, watermark detector (see Fig. 2).

3 Digital Watermarking of Raw Video

The basic idea of watermarking for video is addition of a pseudo-random signal to the video that is below the threshold of perception and that cannot be identified and thus removed without knowledge of the parameters of the watermarking algorithm.

Our approach to accomplish this is a direct extension of ideas from direct-sequence spread spectrum communications.[9] The approach in[3] is similar and was developed independently. Let us denote

$$a_j, \quad a_j \in \{-1, 1\} \tag{1}$$

a sequence of information bits we want to hide in the video stream. We then spread this discrete signal by a large factor cr, called the chip-rate, and obtain the spread sequence

$$b_i = a_j, \quad j \cdot cr \leq i < (j+1) \cdot cr \tag{2}$$

The spread sequence b_i is amplified with an amplitude factor α and modulated with a binary pseudo-noise sequence

$$p_i, \quad p_i \in \{-1, 1\} \tag{3}$$

The modulated signal, i.e. the watermark $w_i = \alpha \cdot b_i \cdot p_i$ is added to the line-scanned digital video signal v_i yielding a watermarked video signal

$$\hat{v}_i = v_i + \alpha \cdot b_i \cdot p_i. \tag{4}$$

Due to the noisy nature of p_i, w_i is also a noise-like signal and thus difficult to detect, locate, and manipulate. The recovery of the hidden information is easily accomplished by multiplying the watermarked video signal with the same pseudo-noise sequence p_i that was used in the coder:

$$s_j = \sum_{i=j\cdot cr}^{(j+1)\cdot cr-1} p_i \cdot \hat{v}_i = \sum_{i=j\cdot cr}^{(j+1)\cdot cr-1} p_i \cdot v_i + \sum_{i=j\cdot cr}^{(j+1)\cdot cr-1} p_i^2 \cdot \alpha \cdot b_i \tag{5}$$

The first term on the right-hand side of (5) vanishes, if

$$\sum_{i=j\cdot cr}^{(j+1)\cdot cr-1} p_i = 0 \tag{6}$$

(i.e., the pseudo-noise sequence contains as many -1's as 1's in the interval $[j \cdot cr \ldots (j+1) \cdot cr))$. p_i and v_i are uncorrelated and therefore $\sum_{i=j \cdot cr}^{(j+1)\cdot cr-1} p_i \cdot v_i = 0$. In practice however, the sum in (6) is not zero, and a correction term

$$\Delta = -(\sum_{i=j\cdot cr}^{(j+1)\cdot cr-1} p_i) \cdot mean(\hat{v}_i), \tag{7}$$

which accounts for the different number of -1's and 1's in the pseudo-noise sequence, has to be added. s_j then ideally becomes

$$s_j = \sum_{i=j\cdot cr}^{(j+1)\cdot cr-1} p_i \cdot \hat{v}_i + \Delta \approx cr \cdot \alpha \cdot a_j \tag{8}$$

and the recovered information bit $\hat{a_j}$ is

$$\hat{a_j} = sign(s_j). \tag{9}$$

A condition for the scheme to work is that for demodulation the same pseudo-noise sequence p_i is used that was used for modulation. Thus, even if the receiver knows the basic scheme, it cannot recover the information without knowledge of the pseudo-noise sequence and its possible shift. For simplicity, we have assumed a binary pseudo-noise sequence in (3). Non-binary PN sequences are also possible without modifications of the scheme, and are in fact favorable in terms of security. Given several sequences with different watermarks, it is easier to figure out the unwatermarked pixel values if the watermark consists only of -1's and 1's. The amplitude factor α can be varied according to local properties of the image and can be used to exploit spatial and temporal masking effects of the human visual system (HVS). Also, an error correcting code can be employed to increase the robustness of the scheme. Several watermarks can be superimposed, if different pseudo-noise sequences are used for modulation. This is due to the fact that different pseudo-noise sequences are in general orthogonal to each other and do not significantly interfere.[9]

4 Digital Watermarking of Compressed Video

In the bitstream domain it is more difficult to embed a watermark into video, especially when the requirement is imposed that the bit-rate may not be increased. MPEG-2 bitstream syntax allows for user data being incorporated into the bitstream (field user_data, can be included in any of sequence, group_of_pictures and picture headers). However, this is not a suitable means of embedding a watermark, since the user data can easily be stripped off the bitstream. Also, adding user data to an MPEG-2 encoded video sequence increases the bit-rate. Again the key idea is to incorporate the watermark into the signal itself, i.e., into the bitstream representing the video frames. In order to understand how we can achieve that we have to take a close look on how a signal block corresponds to the equivalent portion of the bitstream. Let us consider a block of 8×8 samples, originating from a frame of the sequence for I-frames or from a prediction error signal for P- and B-frames, respectively. The block is transformed with the DCT, quantized, zig-zag-scanned and run-level-encoded with VLC codewords for the (run,level)-pairs. Thus, the block of 8×8 samples translates into a codeword representing the DC coefficient followed by a number of VLC codewords representing (run,level)-pairs and hence specifying position and value of one DCT coefficient each. The (run,level)-codewords in MPEG-2 are fixed. Fig. 3 shows the number of bits for the (run,level)-codewords specified in the MPEG-2 VLC tables.[10] (run,level)-combinations that are not specifically represented in the VLC tables are coded with a codeword of 24 bits. In order to add a watermark, we process the encoded video signal block by block. For each signal block, the watermarking procedure consists of the following steps:

1. Calculate the DCT of the watermark (of the spread information bits modulated by the pseudo-noise sequence) for the 8×8-block. Do a zig-zag-scan, yielding a 1×64-vector of re-scanned DCT coefficients. Denote the DCT coefficients by W_n with W_0 being the DC coefficient and W_{63} being the highest-frequent

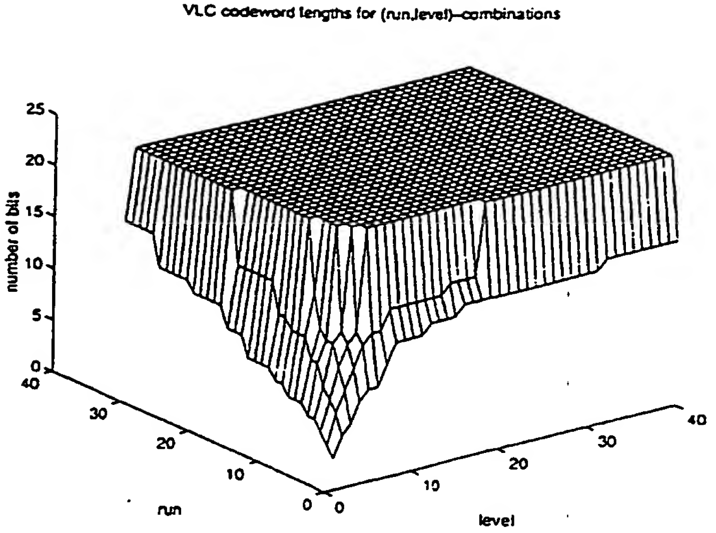

Figure 3: MPEG-2 VLC codeword lengths for (run.level)-codewords.

AC-coefficient. Denote the DCT coefficients of the unwatermarked signal V_n and of the watermarked signal $\tilde{V}_n$.

2. **DC-coefficient**: For the DC-coefficient, $\tilde{V}_0 = V_0 + W_0$, that is, the mean value of the watermark-block is added to the mean value of the signal-block.
3. **AC-coefficients**: Search the bitstream of the coded signal for the next VLC codeword, identify the (run.level)-pair (r_m, l_m) belonging to that codeword and, thus, the position and amplitude of the AC DCT coefficient V_m represented by the VLC codeword.
4. $\tilde{V}_m = V_m + W_m$ is the candidate DCT coefficient for the watermarked signal. However, we do also have the constraint of not increasing the bit-rate. Thus, we have to check the number of bits we have to transmit for the watermarked DCT coefficient $\tilde{V}_m$ versus the bit-rate we have to transmit for the unwatermarked DCT coefficient V_m:
5. Let R be the number of bits used for transmitting the codeword for (r_m, l_m) (i.e., for V_m) and $\tilde{R}$ be the number of bits used for transmitting the codeword for $(r_m, \tilde{l}_m)$ (i.e., for $\tilde{V}_m$). (R and $\tilde{R}$ are determined by the VLC-tables defined in MPEG-2[10]).
6. If the bit-rate shall not be increased and $R \geq \tilde{R}$ (or if the bit-rate of the video may be increased, unconditionally), transmit the codeword for $(r_m, \tilde{l}_m)$. Else, transmit the codeword for (r_m, l_m).
7. Repeat steps 3 to 6 until an end_of_block (EOB) codeword is encountered.

Due to the bit-rate constraint, usually only few DCT coefficients of the watermark can be incorporated per 8 × 8-block, in a lot of cases (especially for coarse quantization) it might be only the DC coefficient as outlined in step 2. As a result, the watermarking scheme in the bitstream domain is less robust than its counterpart in the pixel domain. In other words: in the bitstream domain, only a fraction of the signal energy of the watermark can successfully be embedded. However, for watermarking of video, the chip-rate cr may be chosen to be very high, increasing the robustness to the desired level, but at the same time decreasing the data rate for the watermark.

In practice, step 4 has to be modified in order to avoid drift, which otherwise might occur because we partly alter a previously encoded bitstream. We have to decode the unwatermarked video in parallel and to add not only the watermark, but also to subtract the drift that has been occurred so far.

5 Attacks against Watermarks, and Remedies

One of the main requirements on watermarking schemes is robustness against intentional or unintentional attacks attempting to remove or destroy the watermark. Possible attacks include:

a. Addition of a constant offset

b. Addition of gaussian or non-gaussian noise

c. Linear filtering, e.g. low-pass or high-pass filtering

d. Nonlinear filtering, e.g. median filtering

e. Compression, e.g. by hybrid coding schemes like MPEG or H.263

f. Local exchange of pixels (e.g. permutation of a 2 × 2-block of pixels)

g. Quantization of the pixel gray values

h. Rotation of the video frames

i. Spatial scaling of the video frames

j. Removal or insertion of single pixels

k. Removal or insertion of pixel rows or columns

l. Removal or insertion of video frames

m. Averaging of several versions of the same video with different embedded watermarks

n. Single or multiple analog recording on a VCR

The attacks listed in a.– g. do not pose a real problem to our scheme, if the parameters (especially the chip-rate) are chosen adequately. The same holds for rotation of the video frames (h.), if the rotation angle is very small; otherwise a rotation detection and correction has to be added. Spatial scaling (i.) is critical and a scaling detection and correction mechanism is needed. Removal or insertion of parts of the data (j.–l.) leads to loss of synchronicity of the PN sequence between sender and receiver, and must be considered. A scheme that detects loss of synchronicity and attempts to resynchronize (for example by use of a sliding correlator[9]) must be employed. If complexity has not to be considered, all mentioned attacks can be counter-attacked. A real problem however occurs if several versions of the same video with different embedded watermarks are averaged in order to reconstruct the original pixel values (m.). Countermeasures against this sort of attack are still under research. The effects of analog recording (n.) are typically a combination of the effects mentioned before.

6 Implementation and Simulation Results

We have implemented the outlined scheme as a C program which takes an MPEG-2 bitstream as its input. The program decodes the video and simultaneously parses the bitstream and writes it to a new file. Only those parts of the bitstream containing VLC codewords representing DC- and AC-coefficients of DCT blocks are located and replaced by VLC codewords representing DC- and AC-coefficients of the same block *plus watermark*. Typical parameters are $\alpha = 1 \dots 5$ and $cr = 10,000 \dots 1,000,000$, yielding data rates for the watermark of 1.25...125 bytes/second for NTSC TV resolution. The complexity, as shown in Fig. 4, is much lower than the complexity of a

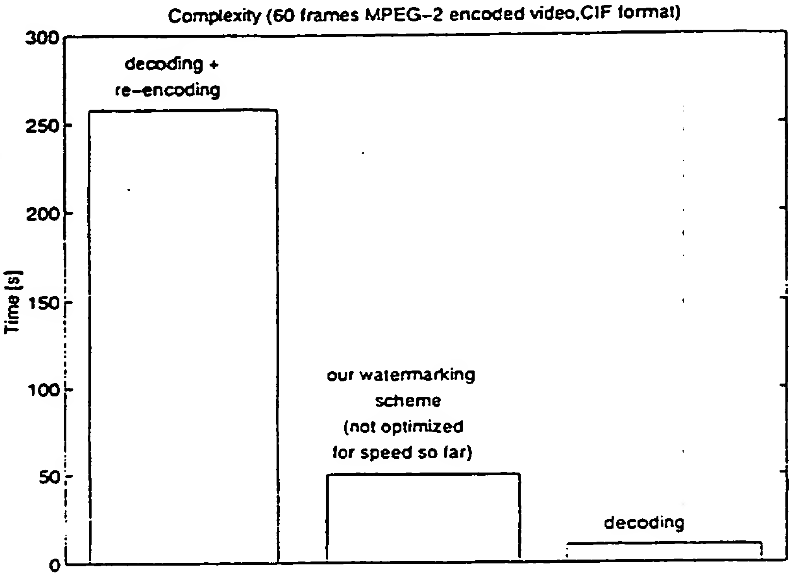

Figure 4: Complexity of our watermarking scheme compared to encoding and decoding.

decoding process followed by watermarking in the pixel domain and re-encoding. For comparison, the complexity of decoding alone is also given. Please note that our program, unlike the public domain MPEG coder and decoder, has not been optimized for speed yet. Figures 5–7 show an example frame from a video sequence. Fig. 5 shows

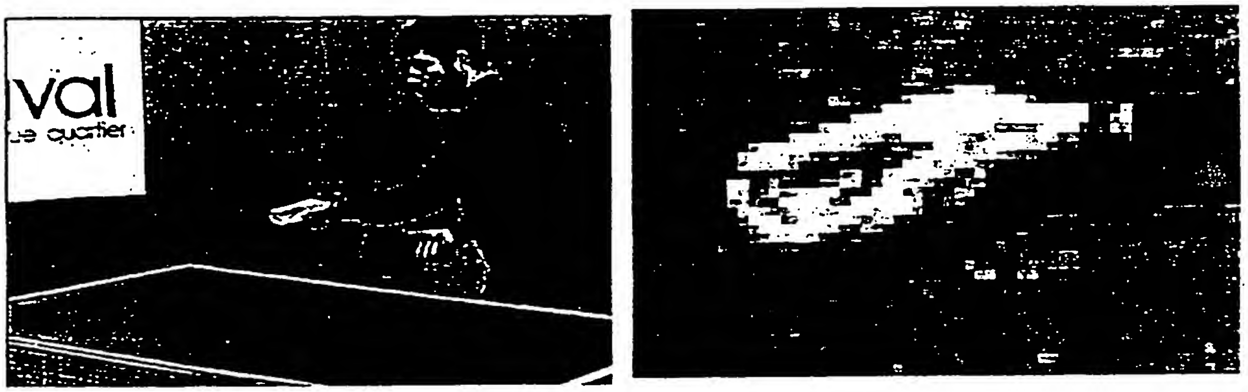

Figure 5: Original

the original frame without compression and a detail from the hand of the table tennis player. Fig. 6 shows the same frame after MPEG-2 encoding and decoding and without an embedded watermark. Fig. 7 finally shows the compressed frame with an embedded watermark. As can be seen, the watermark results in slightly changed pixel amplitudes which are however not visible except in direct comparison to the unwatermarked image. The

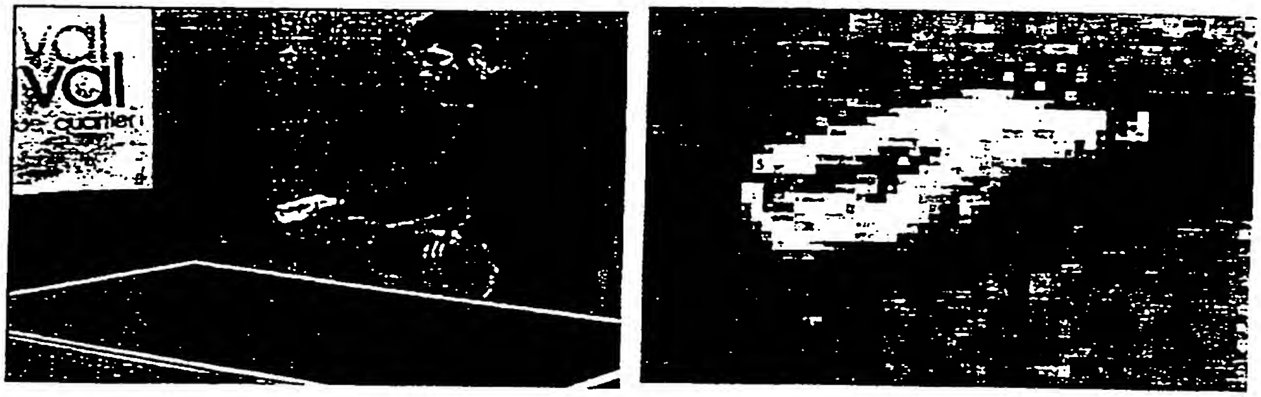

Figure 6: MPEG-2 coded, without watermark

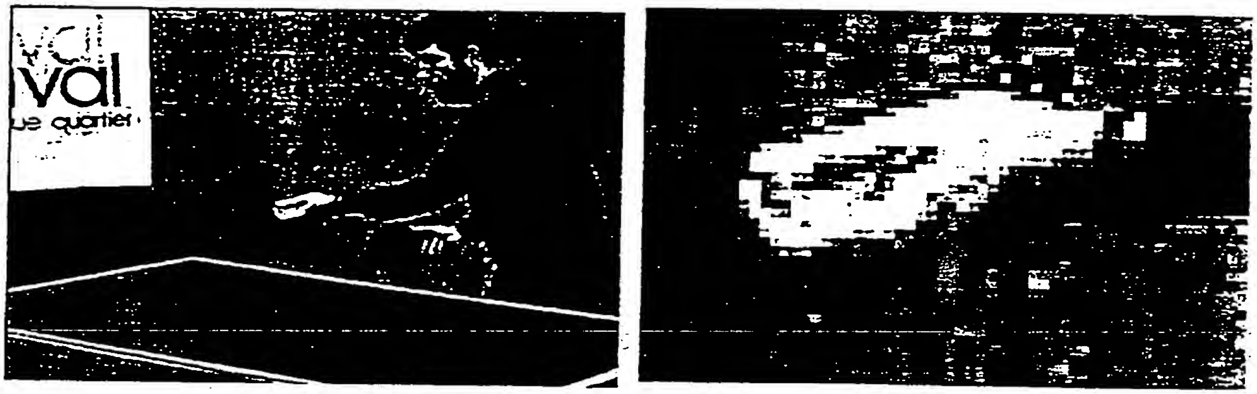

Figure 7: MPEG-2 coded, with embedded watermark

degradation can directly be influenced by varying the amplitude of the watermark. A higher amplitude leads to better robustness, but possibly results in visually annoying distortions.

7 Conclusions

We have presented a novel scheme for watermarking of MPEG-2 compressed video in the bitstream domain. Working on encoded rather than on unencoded video is important for practical watermarking applications. The scheme is interoperable and fully compatible with a scheme working in the pixel domain of uncompressed video which was also presented. With appropriate parameters, the watermarking scheme in the MPEG-2 bitstream domain can achieve netto data rates of several bytes/second while being very robust against unattempted and attempted attacks. The principle can also be applied to other hybrid coding schemes like MPEG-1, ITU-T H.261 or ITU-T H.263.

8 REFERENCES

[1] E. Koch and J. Zhao. Digital copyright labeling: providing evidence of misuse and tracking unauthorized distribution of materials. *OASIS magazine*, December 1995.

[2] E. Koch and J. Zhao. Towards robust and hidden image copyright labeling. In *Proceedings of 1995 IEEE Workshop on Nonlinear Signal and Image processing*, Neos Marmaras, Greece, June 1995.

[3] I. Cox, J. Kilian, T. Leighton, and T. Shamoon. Secure spread spectrum watermarking for multimedia. Technical Report 95-10, NEC Research Institute, Princeton, NJ, USA, 1995.

[4] N. Nikolaidis and I. Pitas. Copyright protection of images using robust digital signatures. In *Proceedings ICASSP 96*, May 1996.

[5] Germano Caronni. Ermitteln unauthorisierter Verteiler von maschinenlesbaren Daten. Technical report, ETH Zürich, Switzerland, August 1993.

[6] Germano Caronni. Assuring ownership rights for digital images. In *Proceedings VIS 95, Session "Reliable IT Systems"*. Vieweg, 1995.

[7] Walter Bender, Daniel Gruhl, and Norishige Morimoto. Techniques for data hiding. Technical report, MIT Media Lab, 1996.

[8] ACCOPI. RACE project M1005 (ACCOPI): Workpackage 8: Watermarking techniques. Technical report, ACCOPI Consortium, April 1995.

[9] David L. Nicholson. *Spread Spectrum Signal Design - Low Probability of Exploitation and Anti-Jam Systems.* Computer Science Press, 1988.

[10] ISO/IEC 13818-2, Generic Coding of Moving Pictures and Associated Audio, Recommendation H.262 (MPEG-2), 1995. International Standard.